

AMENDMENTS TO THE CLAIMS

1. (Currently Amended) A method comprising:
filling a cache line;
receiving a first request for a first segment of the cache line;
indicating at least the first segment is in a ~~non-volatile~~first state that requires that a modification to a segment of a cache line cause a notification of the modification to be sent; and
sending at least the first segment while maintaining a second segment of the cache line in ~~one of a modified volatile state and an exclusive volatile state~~a second state that requires that a modification to a segment of a cache line does not cause a notification of the modification to be sent.
2. (Original) The method of claim 1, further comprising:
modifying at least a portion the first segment of the cache line; and
sending a notification of the modification.
3. (Currently Amended) The method of claim 1, further comprising:
modifying ~~a-~~the second segment of the cache line without generating a notification of the modification; and
indicating the second segment is in a ~~volatile~~second state,
wherein the second state comprises one of: (a) a modified second state that identifies a cache line having a non-volatile segment that is coherent between a plurality of caches associated with different processors, and a second state segment that is not coherent between the plurality of caches, and (b) an exclusive second state that identifies a cache line having a first state segment, a second state segment, and a segment that is owned by a processor other than a processor associate with the cache.
4. (Original) The method of claim 1, wherein the cache line is a part of a first cache associated with a first processor.
5. (Original) The method of claim 4, further comprising:
sending data from the cache line to a second cache associated with a second processor.

6. (Currently Amended) The method of claim 3, further comprising:
receiving a second request for a different third segment of the cache line; and
sending at least the third segment of the cache line while maintaining one of the modified
volatile second state and exclusive volatile second state.

7. (Currently Amended) The method of claim 6, further comprising:
updating the cache line to indicate the third segment of the cache line is in a non-
volatilefirst state.

8. (Currently Amended) The method of claim 6, further comprising:
updating the cache line such that only the third segment of the cache line is in a non-
volatilefirst state; and
invalidating the cache line from all other processors holding the cache line or sending an
updated copy of the cache line to a processor.

9. (Currently Amended) A memory device comprising:
a first plurality of memory segments of a plurality of cache lines to track a volatile second
status for a second plurality of a memory segments of the cache lines, wherein the volatile
second status requires that a modification to a segment of a cache line does not cause a
notification of the modification to be sent, and wherein the second status comprises at least two
of a modified volatilesecond status, a shared volatilesecond status, or an exclusive volatilesecond
status for the second plurality of memory segments,

wherein the modified second status identifies a cache line having a first status segment
that is coherent between a plurality of caches associated with different processors, and a second
status segment that is not coherent between the plurality of caches,

wherein the shared second status identifies a cache line having a first status segment, a
second status segment, and a segment that is owned by a processor other than a processor
associate with the cache, and

wherein the exclusive second status identifies a cache line having a first status segment
that is shared between a plurality of caches associated with different processors and requires that

a modification to a first status segment of a cache line cause a notification of the modification to be sent, and a second status segment that is shared between the plurality of caches; and
circuitry to allow access to the plurality of memory segments.

Claims 10-12 (Canceled).

13. (Currently Amended) A method comprising:
executing a first volatile second state load request for requested data, wherein a second state requires that a modification to a segment of a cache line does not cause a notification of the modification to be sent;

placing the requested data in a segment of a first state cache line, wherein a first state requires that a modification to a segment of a cache line cause a notification of the modification to be sent; and

placing an indication of a shared volatile second state associated with the requested data in the segment of the cache line, wherein the shared second state identifies a cache line having a first state segment that requires that a modification to a first state segment of a cache line cause a notification of the modification to be sent, a second state segment, and a segment that is owned by a processor other than a processor associate with the cache.

14. (Currently Amended) The method of claim 13, further comprising:
executing a load or a second volatile second state load request for data held in the cache line in a non-volatile first state; and
returning the result of the volatile second state load request.

15. (Currently Amended) The method of claim 13, further comprising:
executing a load or second volatile load request for a volatile portion of the cache line and placing the cache line in an invalid state.

16. (Currently Amended) The method of claim 13, further comprising:
executing a load or second volatile second state load request for a volatile second state portion of the cache line and receiving an updated copy of the cache line in a shared volatile second state with requested data in a non-volatile first state.

17. (Currently Amended) An apparatus comprising:
means for storing data; and
means for tracking a shared ~~volatile~~second state, a modified ~~volatile~~second state and an exclusive ~~volatile~~second state of cache line segments for the means for storing data,
wherein the modified second state identifies a cache line having a first state segment that is coherent between a plurality of caches associated with different processors and requires that a modification to a first state segment of a cache line cause a notification of the modification to be sent, and a second state segment that is not coherent between the plurality of caches,
wherein the shared second state identifies a cache line having a first state segment, a second state segment, and a segment that is owned by a processor other than a processor associate with the cache, and
wherein the exclusive second state identifies a cache line having a first state segment that is shared between a plurality of caches associated with different processors, and a second state segment that is shared between the plurality of caches.

18. (Currently Amended) The apparatus of claim 17, further comprising:
means for indicating one of a first portion and a second portion of a segment of the means for storing data contains ~~non-volatile~~first state data, wherein a first state requires that a modification to a segment of a cache line cause a notification of the modification to be sent.

19. (Currently Amended) The apparatus of claim 17, further comprising:
means for notifying a second means for storing data that a ~~non-volatile~~first state data has been modified, wherein a first state requires that a modification to a segment of a cache line cause a notification of the modification to be sent.

20. (Currently Amended) The apparatus of claim 17, further comprising:
means for indicating multiple segments are in one of a ~~volatile~~second state and ~~non-volatile~~a first state for a line of the means for storing data, wherein a first state requires that a modification to a segment of a cache line cause a notification of the modification to be sent, and wherein a second state requires that a modification to a segment of a cache line does not cause a notification of the modification to be sent.

21. (Currently Amended) A system for enabling ~~volatile~~-shared data across caches comprising:

a first cache in a first central processing unit to store a first cache line in one of a shared ~~volatile~~second state, an exclusive ~~volatile~~second state, and a modified ~~volatile~~second state; and a second cache in a second central processing unit in communication via a system interconnect with the first cache to store a second cache line,

wherein the modified second state identifies a cache line having a first state segment that is coherent between a plurality of caches associated with different processors, and a second state segment that is not coherent between the plurality of caches,

wherein the shared second state identifies a cache line having a first state segment that requires that a modification to a first state segment of a cache line cause a notification of the modification to be sent, a second state segment, and a segment that is owned by a processor other than a processor associate with the cache, and

wherein the exclusive second state identifies a cache line having a first state segment that is shared between a plurality of caches associated with different processors, and a second state segment that is shared between the plurality of caches.

22. (Original) The system of claim 21, further comprising:

a first processor associated with the first cache; and
a second processor associated with the second cache.

23. (Original) The system of claim 21, further comprising:
a system memory that is cached by the first and second caches.

24. (Currently Amended) The system of claim 21, wherein the first cache line indicates at least one ~~non-volatile~~first state segment, wherein a first state requires that a modification to a segment of a cache line cause a notification of the modification to be sent.

25. (Currently Amended) The system of claim 21, wherein the first cache notifies the second cache of a change in the ~~non-volatile~~first state portion of a cache line in one of the modified ~~volatile~~second state, the exclusive ~~volatile~~second state, and the shared volatile~~second~~

state, wherein a first state requires that a modification to a segment of a cache line cause a notification of the modification to be sent.

26. (Currently Amended) A processor comprising:

a pipeline to process instructions in one of program order and out of program order;
a set of execution units to execute the instructions; and

a set of caches coupled to the pipeline to store cache line segments of data required by the pipeline in a modified volatilesecond state, an exclusive volatilesecond state, and a shared volatilesecond state,

wherein the modified second state identifies a cache line having a first state segment that is coherent between a plurality of caches associated with different processors, and a second state segment that is not coherent between the plurality of caches,

wherein the shared second state identifies a cache line having a first state segment that requires that a modification to a first state segment of a cache line cause a notification of the modification to be sent, a second state segment, and a segment that is owned by a processor other than a processor associate with the cache, and

wherein the exclusive second state identifies a cache line having a first state segment that is shared between a plurality of caches associated with different processors, and a second state segment that is shared between the plurality of caches.

27. (Currently Amended) The processor of claim 26, wherein the cache generates a notification upon modification of non-volatilefirst state data, wherein a first state requires that a modification to a segment of a cache line cause a notification of the modification to be sent.

28. (Original) The processor of claim 26, wherein the cache shares data containing a modified portion.

29. (Currently Amended) A machine readable medium having instruction stored therein which when executed cause a machine to perform a set of operations comprising:

placing data in a cache line;

indicating the data in the cache line is in one of a modified volatilesecond state, and exclusive volatilesecond state, and a shared volatilesecond state state; and

sharing the data in the cache line,

wherein the modified second state identifies a cache line having a first state segment that is coherent between a plurality of caches associated with different processors, and a second state segment that is not coherent between the plurality of caches,

wherein the shared second state identifies a cache line having a first state segment that requires that a modification to a first state segment of a cache line cause a notification of the modification to be sent, a second state segment, and a segment that is owned by a processor other than a processor associate with the cache, and

wherein the exclusive second state identifies a cache line having a first state segment that is shared between a plurality of caches associated with different processors, and a second state segment that is shared between the plurality of caches.

30. (Currently Amended) The machine readable medium of claim 29, having instructions stored therein which when executed cause a machine to perform a set of operations further comprising:

generating a notification when a ~~non-volatile~~first state data portion is modified, wherein a first state requires that a modification to a segment of a cache line cause a notification of the modification to be sent.

31. (Currently Amended) The machine readable medium of claim 29, having instruction stored therein which when executed cause a machine to perform a set of operations further comprising:

indicating the size and position of a ~~non-volatile~~first state portion of a cache line, wherein a first state requires that a modification to a segment of a cache line cause a notification of the modification to be sent.

32-33. (Canceled).

34. (Currently Amended) The method of claim 32_1, wherein the notification is sent to a processor that: does not own the modified segment, holds the modified segment in a cache

line of a cache associated with the notified processor, or does not hold the modified segment in a cache line of a cache associated with the notified processor.

35. (Currently Amended) The method of claim 32_3, wherein the cache line further comprises:

a lock field, a data field, and a status field, the status field to indicate ~~a volatile~~that the second status ~~comprising~~comprises one of a modified ~~volatile~~second state, a shared ~~volatile~~second state, and an exclusive ~~volatile~~second state.

36. (Currently Amended) The method of claim 32_1, wherein the cache line further comprises a second segment in a ~~volatile~~second state and a third segment in a ~~non-volatile~~first state.

37. (Canceled)

38. (Currently Amended) The method of claim 13, wherein the second state comprises:

a modified volatile state that identifies a cache line having a ~~non-volatile~~first state segment that is coherent between a plurality of caches associated with different processors, and a ~~volatile~~second state segment that is not coherent between the plurality of caches; and

an exclusive volatile state that identifies a cache line having a ~~non-volatile~~first state segment, a ~~volatile~~second state segment, and a segment that is owned by a processor other than a processor associated with the cache; and

~~a shared volatile state identifies a cache line having a non volatile segment that is shared between a plurality of caches associated with different processors, and a volatile segment that is shared between the plurality of caches.~~

39. (Currently Amended) The method of claim 38, wherein the cache line further comprises:

a lock field, a data field, and a status field, the status field to indicate ~~a volatile~~that the second status ~~comprising~~comprises one of a modified ~~volatile~~second state, a shared ~~volatile~~second state, and an exclusive ~~volatile~~second state.

40. (Currently Amended) The method of claim 38, wherein the segment is a first segment; and wherein the cache line further comprises a second segment in a ~~volatile~~second state and a third segment in a ~~non-volatile~~first state, ~~a non-volatile state requires that a modification to a segment of a cache line cause a notification of the modification to be sent, and a volatile state requires that a modification to a segment of a cache line does not cause a notification of the modification to be sent.~~

41. (Canceled)

42. (Currently Amended) The apparatus of claim 41-17, further comprising: means for tracking a lock field and a data field for the means for storing data.

43. (Currently Amended) The apparatus of claim 41-17, further comprising: means for tracking a ~~volatile~~second state and a ~~non-volatile~~first state for the means for storing data, a ~~non-volatile~~first state requires that a modification to a segment of a cache line cause a notification of the modification to be sent, and a ~~volatile~~second state requires that a modification to a segment of a cache line does not cause a notification of the modification to be sent.

44-46. (Canceled).